

(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
PARIS

(11) N° de publication :

(à n'utiliser que pour les  
commandes de reproduction)

2 798 205

(21) N° d'enregistrement national :

00 08787

(51) Int Cl<sup>7</sup> : G 06 F 13/00, G 11 C 16/10

(12)

## DEMANDE DE BREVET D'INVENTION

A1

(22) Date de dépôt : 06.07.00.

(30) Priorité : 07.07.99 DE 19931184.

(71) Demandeur(s) : ROBERT BOSCH GMBH Gesellschaft  
mit beschränkter Haftung — DE.

(43) Date de mise à la disposition du public de la  
demande : 09.03.01 Bulletin 01/10.

(72) Inventeur(s) : HURICH MARTIN, SEYFANG  
ROBERT et TENBUSCH EBERHARD.

(56) Liste des documents cités dans le rapport de  
recherche préliminaire : Ce dernier n'a pas été  
établi à la date de publication de la demande.

(60) Références à d'autres documents nationaux  
apparentés :

(73) Titulaire(s) :

(54) PROCEDE ET DISPOSITIF POUR MODIFIER LE CONTENU DE LA MEMOIRE D'APPAREIL DE COMMANDE  
NOTAMMENT POUR LES OPERATIONS DE FONCTIONNEMENT D'UN VEHICULE.

(57) Procédé et dispositif pour modifier et/ ou enregistrer  
des données et/ ou des programmes dans une première  
mémoire non fugitive d'un système d'ordinateur notamment  
une mémoire d'une unité de commande d'un véhicule. Les  
données et/ ou programmes sont répartis en première et se-  
conde données et/ ou programmes; les secondes données  
et/ ou programmes méritent d'être protégés ou contiennent  
des données et/ ou programmes le méritant. Les anciennes  
secondes données et/ ou programmes dans la zone origina-  
le associée sont d'abord effacés si des nouvelles secondes  
données et/ ou programmes sont enregistrés dans une se-  
conde zone de mémoire ne correspondant pas à la zone ori-  
ginale et un programme de copie est enregistré dans la  
seconde mémoire fugitive. Le programme de copie inscrit  
les nouvelles secondes données et/ ou programmes à partir  
de la seconde zone de mémoire dans la zone originale. Il  
n'est pas nécessaire de faire une copie de sécurité. Un or-  
dre d'effacement émis par l'appareil de programmation n'ef-  
face pas dans ces conditions les anciennes secondes  
données et/ ou programmes et les reprend pour la coman-  
de de déroulement.



**Etat de la technique**

La présente invention concerne un procédé et un dispositif pour modifier et/ou enregistrer des données et/ou des programmes, notamment des données et/ou des programmes de commande des opérations de fonctionnement dans un véhicule, dans au moins une mémoire non fugitive d'un système d'ordinateur, notamment une mémoire d'une unité de commande d'un véhicule, selon lequel, déjà dans la mémoire non fugitive, dans une zone originale, associée, on efface les anciennes données et/ou programmes avant de modifier et/ou d'enregistrer des données et/ou programmes nouveaux.

Les données et/ou programmes sont modifiés et/ou enregistrés dans au moins une mémoire d'un système d'ordinateur, notamment d'une unité de commande des opérations de fonctionnement d'un véhicule.

Selon le document DE 43 32 499 A1, on connaît un procédé pour reprogrammer complètement une mémoire non fugitive, effaçable électriquement équipant un appareil de commande, notamment un appareil de commande de véhicule automobile. L'opération de programmation se déroule ainsi sans utiliser une mémoire non modifiable (ROM). La mémoire effaçable, non fugitive est répartie en au moins deux zones distinctes effaçables et programmables dans lesquelles on inscrit chaque fois un module de programmation. En répartissant deux zones, il est possible en cas de coupure de tension, avant la fin de l'opération de programmation, de répéter complètement l'opération de programmation entre le véhicule automobile et le support de données car le module de programmation est quasiment dédoublé dans la mémoire programmable non fugitive, c'est-à-dire que cela existe dans les deux zones de mémoire. Ainsi, même en cas d'effacement de l'un des deux modules de programmation, on peut néanmoins avoir un module de programmation intact comme copie de sécurité dans une mémoire. Pour cela, il faut que les copies de sécurité soient également déposées de manière continue car une perturbation comme par exemple une chute de tension nécessite une nouvelle programmation complète. Cette nouvelle

programmation se déroule alors non pas de manière automatique, mais elle doit être prévue de l'extérieur.

On a constaté que les procédés ci-dessus ne permettent pas d'obtenir toujours des résultats optimum. C'est ainsi qu'il n'y a pas de rétablissement automatique de zones incomplètes résultant d'une destruction et qu'il faut reprogrammer complètement à partir de l'extérieur. Dans ces conditions, dans les procédés connus, on enregistre des copies de sécurité des données des programmes, de manière permanente dans une zone de sauvegarde. Les données et/ou programmes méritant d'être protégés sont ainsi dupliqués par des sauvegardes, c'est-à-dire des copies de sécurité qui doivent pouvoir fonctionner. Cela donne plusieurs cycles d'effacement et de programmation qui chargent le composant contenant la mémoire. En général, selon l'état de la technique, il faut un grand nombre d'états d'ordre pour reprogrammer une mémoire. Cela fait que l'on ne dispose pas d'un code de programme fonctionnel relativement longtemps notamment dans les phases de copie de la sauvegarde.

En arrêtant la routine de copie lors de la reprogrammation, on utilise ensuite une sauvegarde avec toutes les anciennes données et/ou programmes ou routines. Comme la sauvegarde utilisée à cet effet comprend toutes les données et/ou programmes méritant d'être protégés, la taille du code qu'il faut exécuter en cas de coupure est relativement importante. C'est pourquoi, dans l'état de la technique, suivant l'importance de la zone des données et/ou programmes méritant d'être protégés, il faut une mémoire de travail relativement importante (RAM) par exemple pour y exécuter la routine de programmation de cette zone méritant d'être protégée. Les blocs Boot variables ainsi utilisés comme environnement de programmation contenant les programmes de démarrage du système d'ordinateur, décident s'il faut précisément exécuter les anciennes ou les nouvelles données et/ou zones de programmes sont des sources de nouvelles erreurs.

Avant qu'un programme ou sa copie de sécurité ne soit déroulé, on peut tester les éventuels défauts de données dans le code programme. Pour reconnaître les erreurs de don-

nées par exemple en cas d'enregistrement en mémoire ou de programmation défectueuse, on connaît de nombreux procédés. Il faut que les données et/ou programmes soient représentés dans un code d'erreurs ou un code de reconnaissance d'erreurs 5 dans lequel à côté des informations proprement dites, il y a également les informations redondantes pour la reconnaissance des erreurs. De telles informations sont par exemple les bits de parité qui permettent une reconnaissance de défauts en formant des sommes de contrôle telles que des sommes trans- 10 versales et/ou des sommes longitudinales (contrôle des bits de parité). D'autres procédés de contrôle exécutables de manière linéaire et/ou cyclique sont ceux du code Hamming, du code BCH (Bose, Chandhuri, Hocquenghem) et du code Abramson. Ces procédés de vérification englobent également le contrôle 15 de redondance cyclique (CRC) selon lequel à partir des données et/ou des programmes, c'est-à-dire des informations à effacer et/ou à copier et/ou à enregistrer, on génère par blocs des octets de contrôle précisément selon le procédé CRC et on traite les informations ou on les enregistre. Dans 20 quelques-uns de ces procédés, on prévoit une correction des erreurs en plus de leur détection.

#### **Avantages de l'invention**

Pour remédier aux inconvénients ci-dessus, selon l'invention, on crée un procédé et un dispositif utilisant le 25 procédé, caractérisés en ce que :

- les données et/ou programmes sont répartis en au moins des premières et des secondes données et/ou programmes, les secondes données ou programmes méritant d'être protégés ou contenant des données et/ou programmes méritant d'être 30 protégés,
- les anciennes secondes données et/ou programmes ne sont d'abord effacés de la zone originale associée que lorsqu'on inscrit les nouvelles secondes données et/ou programmes dans une seconde zone de mémoire qui ne correspond 35 pas à la zone originale, et
- on inscrit un programme de copie dans une mémoire fugitive qui peut alors copier les nouvelles secondes données et/ou

programmes de la seconde zone de mémoire dans la zone originale.

Selon une autre caractéristique, on développe un procédé et un dispositif utilisant ce procédé, caractérisés en ce qu'un programme de copie inscrit dans une mémoire fugitive enregistre les nouvelles données et/ou programmes dans la zone de mémoire, après avoir effacé les anciennes données et/ou programmes sans réaliser une copie de sécurité des anciennes données et/ou programmes.

Pour ces procédés ci-dessus, de façon avantageuse, selon l'invention on peut prévoir que le programme de copie soit contenu dans les secondes données et/ou programmes et puisse être chargé par ceux-ci dans la mémoire fugitive. Mais on peut aussi prévoir que le programme de copie est conservé dans des troisièmes données et/ou programmes qui peuvent exécuter une opération de démarrage du système d'ordinateur, et peut être chargé dans la mémoire fugitive par les troisièmes données et/ou programmes.

Selon une autre caractéristique de l'invention, on efface seulement partiellement les anciennes données et/ou programmes par l'ordre d'effacement, les données et/ou les programmes étant répartis en au moins des premières et des secondes données et/ou programmes, au moins les secondes données et/ou programmes méritant d'être protégés, ou contenant des données et/ou des programmes méritant d'être protégés, et les anciennes secondes données et/ou programmes dans la zone de mémoire associée restant conservés dans la zone originale.

Dans cette dernière solution, comme aussi dans la première, on peut prévoir que les secondes données et/ou programmes ne sont ou ne peuvent être exécutés que dans la zone originale.

Selon l'invention, de manière avantageuse, la zone originale des données et/ou programmes méritant d'être protégés n'est pas effacée à l'arrivée d'un ordre d'effacement. Le code programme reste conservé pour la reprogrammation. Au lieu de cela, on efface la zone qui contiendrait normalement la copie des données et/ou programmes méritant d'être protégés, c'est-à-dire la sauvegarde. Cette zone est

utilisée selon une autre façon avantageuse comme mémoire intermédiaire pour les nouvelles données et/ou programmes méritant d'être protégés et on peut renoncer à ce que ces éléments puissent fonctionner. L'expression donnée et/ou programmes méritant d'être protégés, désigne des éléments dont la perte ou les défauts inhérents ne permettent plus une nouvelle programmation de la mémoire ou une modification future du contenu de la mémoire ou de l'utilisation prévue du système d'ordinateur, par exemple pour commander les opérations de fonctionnement dans un véhicule.

La partie du flux de données envisagée pour la zone méritant d'être protégée, selon la programmation est déviée avantageusement dans la zone de mémoire de données utilisée comme mémoire intermédiaire et qui a été préalablement effacée à cet effet.

Par une petite routine optimale (routine RAM, RR) qui se trouve dans la mémoire de travail, on copie alors la nouvelle zone de la mémoire intermédiaire par-dessus l'ancienne zone préalablement non effacée. La mémoire intermédiaire est alors effacée finalement par exemple par la routine RAM, RR ou un programme des nouvelles données et/ou programmes méritant d'être protégés. Finalement, cela permet d'enregistrer dans la zone de mémoire utilisée comme mémoire intermédiaire, d'autres données et/ou programmes.

De façon avantageuse, le procédé ne nécessite qu'un seul ordre d'effacement pour la zone à effacer qui lance le procédé. L'ordre d'effacement est ainsi indépendant de ce que des données méritant d'être protégées sont contenues ou non dans la zone à effacer. La programmation du nouveau code de programme ou des nouvelles données se fait ainsi automatiquement au cours d'une seule étape.

De manière avantageuse, la zone de mémoire méritant d'être protégée est disponible pendant tout le temps de la reprogrammation, ou encore le temps au cours duquel aucun code de programme fonctionnant dans la zone méritant d'être protégée n'est disponible, est réduit au minimum, car les données et/ou programmes méritant d'être protégés se trouvent toujours dans la zone de mémoire correspondante.

Il est en outre avantageux que les routines de programmation complexe des données et/ou programmes méritant d'être protégés ne soient pas enregistrées dans la mémoire de travail. Seule la petite routine RAM, optimisée qui se trouve 5 dans la mémoire de travail, copie les nouvelles données et/ou programmes méritant d'être protégés par-dessus les anciennes informations. Pour cette raison, de façon avantageuse, on peut utiliser une petite mémoire de travail ou encore la charge appliquée à la mémoire de travail est faible. Si pendant 10 l'activité de la routine RAM il y a une coupure, un bloc Boot, conçu de manière invariable peut compléter cette opération à l'aide d'une routine de copie, qui lui est propre. Ainsi, de manière avantageuse, on minimise la taille du code programme à exécuter en cas de coupure.

15 Il est en outre avantageux que le bloc Boot dans la zone de mémoire non effaçable ou non inscriptible ne décide pas de la zone de mémoire qui doit être exécutée, c'est-à-dire avec les données et/ou programmes anciens ou nouveaux, mais le cas échéant il copie le nouveau code du programme 20 par-dessus l'original ancien et exécute celui-ci de nouveau dans la zone originale. Ainsi, de manière avantageuse, les routines ou programmes des données et/ou programmes méritant d'être protégés se trouvent à tout instant dans les zones de mémoire originale qui leur étaient attribuées à l'origine. On 25 réduit ainsi au minimum les risques d'erreurs. Cette procédure ne nécessite pas de sauvegarde susceptible de fonctionner. Cela signifie que même en cas de coupure de programmation, de manière avantageuse, il n'est pas nécessaire d'utiliser une sauvegarde d'anciennes données et/ou programmes mais on peut utiliser les nouvelles données et/ou programmes existants par le bloc Boot vers la zone de destination en les copiant dans la zone originale.

#### Dessins

La présente invention sera décrite ci-après de 35 manière plus détaillée à l'aide des dessins annexés dans lesquels :

- la figure 1 montre le matériel selon l'invention,

- la figure 2 montre une répartition possible des mémoires en différentes zones de mémoire,
- la figure 3 représente un ordinogramme du procédé de modification et/ou de programmation pour des programmes et/ou des données,
- 5 - la figure 4 montre la modification et/ou la programmation d'autres données.

#### **Description de l'exemple de réalisation**

La figure 1 montre un exemple de circuit câblé pour convertir le procédé présenté. La référence 100 désigne 10 un système d'ordinateur, notamment un appareil de commande. Les références 104, 105 et en option 106 correspondent à différentes mémoires. La référence 104 représente une mémoire effaçable non fugitive par exemple une mémoire Flash-EPROM, 15 permettant un effacement et une programmation par zones ou par blocs du contenu de la mémoire. Il existe de plus une mémoire fugitive 105 par exemple sous la forme d'une mémoire RAM, constituant une seconde mémoire servant entre autres à enregistrer de façon intermédiaire des données et/ou des programmes. Dans cet exemple de réalisation, on enregistre le 20 programme de copie ou la routine RAM, RR, optimisé. La mémoire RAM 105 peut également être démarrée par des routines de programme. En option, on peut prévoir une autre mémoire non fugitive 106. Les mémoires 104, 105 et en option 106 sont 25 reliées par un système de bus 103 entre autres avec un processeur CPU 101 et une unité entrée/sortie 102. La mémoire 106 peut également être une mémoire externe supplémentaire en dehors du système interne du micro-ordinateur ou même à l'extérieur du système d'ordinateur. De plus, le système d'ordinateur 30 peut comporter notamment un appareil de commande ainsi que d'autres composants non représentés ici comme par exemple d'autres processeurs CPU et mémoire, diverses unités entrée/sortie, etc.. Une interface entrée/sortie 107 par exemple de type série ou parallèle relie le système d'ordinateur 35 100 à un appareil de commande ou de programmation 108 externe. Cet appareil de programmation externe 108 qui peut également être un ordinateur usuel, sert à l'entrée de données et/ou de programmes par exemple pour programmer une nou-

velle version de programmes et/ou de données. De plus, à cet endroit, il y a également d'autres moyens d'entrée qui peuvent par exemple être reliés directement aux autres unités entrée/sortie.

5 La répartition des mémoires ci-dessus en zone de mémoire est représentée à la figure 2. Selon cette figure, la référence 200 désigne symboliquement l'ensemble de la mémoire Flash-EPROM 104. Celle-ci contient entre autres une zone de mémoire SB 201 qui représente l'environnement de programmation correspondant sous la forme de blocs de lancement, c'est-à-dire entre autres le programme de lancement du système d'ordinateur. La zone de mémoire SB 201 pour le bloc BB peut être conçue pour n'autoriser ni la lecture ni l'enregistrement dans cette zone car le bloc de lancement doit rester 10 inchangé. Les programmes de démarrage sont lancés lors du démarrage normal du système ainsi qu'après une remise à l'état initial à cause d'une coupure de tension ou pour une autre origine. Une zone de programme pour les premiers programmes P1 est représentée avec la zone SB 202. Il existe à côté une 15 zone de mémoire SB 203, une zone de données qui contient la fin d'une opération de programmation, d'autres données D.

20

De plus, avec la zone de mémoire SB 204, on a représenté la zone dans laquelle sont enregistrées les données et/ou les programmes méritant d'être protégés ou des données et/ou des programmes méritant d'être protégés. Dans un but de simplification, les données enregistrées dans SB 204 sont désignées de manière générale sous le terme de données et/ou de programmes méritant d'être protégés SDP. En particulier, lors 25 de l'utilisation du procédé et du dispositif présenté, pour la commande des opérations de fonctionnement se déroulant dans un véhicule, cette zone SB 204 se trouve par exemple dans la zone de mémoire dans laquelle est enregistré le programme de roulement. La zone de mémoire SB 204 est ainsi la zone originale de SDP. Les routines spécifiques nécessaires à 30 la programmation Flash, entre autres contenues dans les données et/ou les programmes méritant d'être protégés SDP, doivent être disponibles pendant toute la durée du changement et/ou de la programmation. De plus, les données et/ou les 35

programmes méritant d'être protégés SDB, ne peuvent être transformés par programmes, c'est-à-dire changés pour ne pas nécessiter de compatibilité descendante des interfaces entre par exemple les programmes de roulement et les informations SDP. L'importance donnée par le dessin aux zones de mémoire est arbitraire et ne correspond pas à la taille réelle car celle-ci est différente selon le domaine d'utilisation et l'application. Ainsi, les zones de mémoire SB 201 - SB 204 peuvent remplir toute la mémoire Flash 200 représentée symboliquement. Par les données et/ou programmes méritant d'être protégés SDP, enregistrés dans la zone de mémoire 204, on définit un environnement de programme propre. La zone de mémoire SB 204 correspond comme indiqué à la zone originale pour SDP. Dans la mémoire Flash 200 représentée symboliquement, il est prévu une zone de sauvegarde, propre, réservée en permanence ni pour les programmes ni pour les données notamment pas pour les données et/ou programmes méritant d'être protégés SDP.

En option, comme le montre la figure 1, on peut utiliser un autre élément de mémoire, notamment une mémoire Flash/EPROM. La mémoire en option, supplémentaire 106 est représentée symboliquement par le bloc 208. La zone 209 peut correspondre à une zone pour d'autres programmes et/ou données. Dans l'exemple décrit, on a enregistré un second programme P2. Cela peut par exemple être nécessaire si la zone de mémoire pour les premiers programmes P1 pour le code programme nécessaire composé de P1 et de P2 n'est pas suffisante. On peut également envisager de réaliser par les programmes P1, certaines fonctions standards et/ou de base de l'application respective et de représenter les autres variations comme des modes de réalisation spéciaux, des équipements particuliers comme par exemple pour des véhicules automobiles ainsi que des fonctions spéciales par les programmes P2 dans la zone de mémoire SB 209.

La mémoire de travail ou mémoire RAM 105 selon la figure 1 est représentée symboliquement par le bloc 206. Cette mémoire contient une zone de mémoire 207 pour la partie contenant la routine RAM optimisée RR, qui peut par exemple

être chargée dans la mémoire de travail également par n'importe quelle zone de programmes comme le bloc Boot BB, des zones de programmes P1 ou P2 ainsi que les données et/ou programmes méritant d'être protégés SDP. Cela dépend de la zone 5 dans laquelle est localisée la fonctionnalité nécessaire par exemple une fonction de copie pour le transfert. La fonction de copie qui exécute le transfert de la routine RAM, RR dans la mémoire de travail 206 ou dans la zone de mémoire 207 connaît l'adresse initiale et l'adresse finale de la mémoire 10 RAM, RR et de plus l'adresse de destination dans RAM. Cette fonction de copie est par exemple une boucle de programmes dans laquelle on incrémente un index source et un index destination en commençant aux adresses de départ respectives jusqu'à atteindre l'adresse finale de la source de l'opération 15 de copie. A chaque adresse qui passe, on peut exécuter alors un transfert de données.

La routine RAM, RR se situe par exemple dans la zone des données et/ou programmes méritant d'être protégés SDP et/ou dans le bloc Boot BB. La routine RAM, RR est par 20 exemple copiée par le programme exécutable dans SDP, vers la mémoire de travail 206 pour y être exécutée. Lorsqu'une opération de copie a été interrompue pour une raison quelconque par RR, on peut charger cette routine RR du bloc Boot BB dans la mémoire de travail 206 et l'exécuter.

25 Le bloc Boot BB se trouve soit dans une zone inaccessible pour l'utilisateur et protégée contre l'inscription dans la mémoire Flash 104 ou 200 ou encore directement localisé dans le processeur CPU 101. De manière avantageuse, la routine RAM optimisée RR, comme indiqué, peut être copiée 30 par le bloc Boot BB par exemple après une défaillance du système ou autre coupure dans la mémoire de travail.

Les programmes et/ou les boucles de programmes et/ou les routines à exécuter dans le procédé sont avantageusement réalisés sous la forme de codes de programmation dans 35 une mémoire. Une réalisation de la combinaison mémoire-programme en logique câblée peut également s'envisager.

Le concept de procédé représenté dans les ordigrammes des figures 3 et 4 contient en liaison avec les zones

de mémoire évoquées ci-dessus, le fait que la zone originale des données et/ou programmes méritant d'être protégés SDP, c'est-à-dire la zone de mémoire SB 204 ne soit pas effacée à la réception d'un ordre d'effacement de l'ensemble de la mémoire 104 ou de parties de la mémoire 104 ou 200. Ainsi, le code de programmes dans la zone SDP, c'est-à-dire SB 204 reste tout d'abord conservé. Au lieu de cela, on efface la zone c'est-à-dire SB 203 qui contiendrait normalement la sauvegarde, dans un autre cas la zone de données pour les données ne méritant pas d'être protégées D.

Pour la programmation suivante, on dévie la partie envisagée du flux de données pour la zone méritant d'être protégée SB 204, c'est-à-dire SDP, vers la zone SB 203 préalablement effacée. Les données et/ou programmes méritant d'être protégés SDP ne doivent pas pouvoir être exécutés dans SB 203. Cela permet d'utiliser des programmes SDP et/ou des données d'origine quelconque, qui doivent uniquement être conçus pour être exécutés dans SB 204.

Par une petite routine optimisée RR, placée dans la mémoire dans la zone SB 207, on copie le contenu de SB 203, c'est-à-dire les données et/ou programmes nouveaux méritant d'être protégés SDP<sub>nouv</sub>, déviés, alors par-dessus les données SDP<sub>anc</sub> qui se trouvent dans la zone SB 204, et ensuite on efface la zone de mémoire SB 203. La zone de mémoire SB 203 sert ainsi uniquement de mémoire intermédiaire pour les données et/ou programmes nouveaux méritant d'être protégés SDP<sub>nouv</sub>.

Si pendant l'activité de la routine RAM, RR, il y a une coupure, on peut terminer l'opération par un bloc Boot BB inchangé dans la zone de mémoire SB 201 à l'aide de sa propre routine de copie. Cette routine de copie est comparable à la routine RAM, RR.

Les figures 3 et 4 donnent une vue d'ensemble du déroulement du procédé. A la figure 3, on programme d'abord les programmes et les données. La zone de programme PB comprend les programmes P1, les données et/ou programmes méritant d'être protégés SDP ainsi qu'en option les programmes P2. Les éléments P1, SDP et P2 en option sont programmés dans

le cadre d'une cession de programmation. L'autre plage de données DB est également programmée dans le cadre d'une cession de programmation de données qui fait directement suite à la cession de programmation de programmes, de manière automatique. Le bloc Boot BB est par exemple conçu pour n'être ni effaçable ni programmable et c'est pourquoi, il est présent en permanence. Dans le bloc 300 on démarre le déroulement du procédé par exemple par l'introduction d'un ordre d'effacement d'une installation de programmation externe pour une certaine plage de mémoires ou pour l'ensemble de la mémoire Flash. Si le bloc Boot BB est conçu invariable et s'il n'est contenu ni dans la mémoire Flash ni par exemple dans le processeur CPU, ce bloc reste naturellement conservé.

Dans le bloc 301, on efface tout d'abord la zone de programmes PB. Comme déjà indiqué, la zone SB 204 avec SDPanc n'est pas effacée. Ainsi, les programmes existant maintenant P1anc dans SB 202 et en option P2anc dans SB 209 sont effacés. Les données D sous la forme de Danc peuvent maintenant également être effacées car dans ce mode de réalisation de la programmation de zones de programmes et de données, il vient nécessairement ensuite une programmation de données. Ainsi, dans le bloc 301, on efface également Danc dans la zone de mémoire SB 203. Le point de départ a été l'émission d'un seul ordre d'effacement pour l'ensemble de la zone à effacer, notamment toute la mémoire Flash 200. Cela est indépendant de ce que la zone à effacer ou à programmer est simplement liée ou s'il s'agit pour la zone à effacer ou à programmer, de zones inférieures méritant d'être protégées et qui nécessitent un traitement particulier.

Après l'effacement on arrive au bloc 302. Dans celui-ci on programme la zone de programmes PB. Les nouveaux programmes P1nouv et en option P2nouv sont alors inscrits dans les zones de mémoire respectives SB 202, SB 209. Les données et/ou programmes nouveaux, méritant d'être protégés SDPnouv ne sont pas inscrits dans SB 204 mais dans la zone de données effacée SB 203 et peuvent ainsi commander la suite du déroulement de SDPanc dans SB 204.

Après la programmation de la zone de programmes PB, comme décrit, on arrive au bloc 303 dans lequel on vérifie la zone de programmes PB. On peut contrôler différents procédés déjà évoqués dans l'état de la technique. Ainsi, on vérifie  $P1_{nouv}$  dans SB 202, en option  $P2_{nouv}$  dans SB 209 et finalement  $SDP_{nouv}$  dans la zone de données SB 203 en utilisant pour le contrôle les données et/ou programmes méritant d'être protégés,  $SDP_{anc}$ , enregistrés encore dans SB 204 avant que  $SDP_{nouv}$  reprenne la commande de déroulement après avoir été copié en retour dans SB 204.

Dans l'interrogation 304, on vérifie si des défauts ne sont produits lors du contrôle. Dans l'affirmative, on revient au bloc 301 et la zone de programmes PB est de nouveau effacée. Cette opération est toujours encore commandée par  $SDP_{anc}$ . Si lors du contrôle on n'a pas trouvé de défauts ou si on a pu les corriger, on passe au bloc 305.

Dans ce bloc on efface la zone de données DB. Tout d'abord dans le bloc 305, on copie la routine RAM, RR dans la zone RAM, SB 207. Cette routine RR peut par exemple être contenue dans  $SDP_{anc}$  dans SB 204 ou par copie de parties de  $SDP_{anc}$  dans la mémoire RAM 206. La routine RAM, RR dans SB 207 efface les données et/ou programmes méritant d'être protégés  $SDP_{anc}$  dans SB 204. Puis, également avec la routine RAM, RR, on copie les données et/ou programmes, nouveaux, méritant d'être protégés, qui viennent d'être contrôlés,  $SDP_{nouv}$  de la zone de données SB 203 dans la zone SB 204. Ensuite, on efface  $SDP_{nouv}$  dans la zone de mémoire SB 203. Ainsi, dans le bloc 305, la commande de déroulement est quasi toujours encore sur  $SDP_{anc}$  d'où on forme ou on déplace la routine RAM, RR. Puis on arrive au bloc 306.

Dans ce bloc on programme la zone de données DB. Les nouvelles données  $D_{nouv}$  sont enregistrées dans SB 203 d'où les données  $SDP_{nouv}$  ont été copiées par  $SDP_{anc}$  dans SB 204. Dans ce bloc on reprend ainsi la commande de déroulement de  $SDP_{nouv}$  dans la zone de mémoire SB 204.

Dans le bloc suivant 307, on vérifie alors la zone de données DB et on vérifie les données  $D_{nouv}$  dans la zone de mémoire SB 203. Cela peut également se faire comme

pour la zone de programmes par les mesures évoquées dans l'état de la technique. Si le contrôle est en ordre, s'il n'y a pas de défauts ou si ceux-ci sont corrigés, on arrive au bloc 309 à la fin du déroulement. En cas de défauts non corrigés ou qui ne peuvent être corrigés, l'interrogation 308 indique que la zone de données n'a pas été programmée correctement et on revient au bloc 301 et l'opération décrite ci-dessus recommence à zéro.

Ainsi, SDP<sub>anc</sub> dans SB 204 n'est pas effacé jusqu'à ce que le contrôle de SDP<sub>nouv</sub> dans SB 203 soit réussi. Cela garantit que même pour des coupures de programmation ou des essais de manipulation, on dispose toujours d'une variante vérifiée des données et/ou programmes méritant d'être protégés SDP dans l'appareil de commande, garantissant une possibilité de reprogrammation permanente.

Dans le déroulement exposé, le code programme de SDP est toujours exécuté à partir de la zone de mémoire SB 204 et non à partir de sa zone de sauvegarde possible, la zone de données SB 203. Ce principe est beaucoup plus sûr avec une mise en œuvre de moyens comparativement faibles pour le développement du programme. Le procédé est ainsi relativement simple et nécessite uniquement en liaison avec la programmation par exemple d'un programme de véhicule pour un véhicule, un ordre suivant ou un retard de l'ordre de programmation qui se termine pendant le traitement de la copie de la nouvelle partie de programme dans la zone de destination, voire la zone originale. Cela est envisageable de manière générale.

Pour les installations de programmation possibles, la répartition de mémoire est avantageusement inconnue. Ces installations fournissent chaque fois un ordre d'effacement à la plage de programmes PB et/ou à la zone de données DB. La déviation de SDP et la recopie de la zone temporaire SB 203 dans la zone de destination proprement dite SB 204 sont par exemple couplées aux opérations d'effacement et n'apparaissent pas pour les installations de programmation possibles. Les installations de programmation, pour modifier et/ou enregistrer des données et/ou des programmes peuvent

être externes vis-à-vis de l'unité de commande contenant la mémoire, comme par exemple l'appareil de programmation 108 de la figure 1, ou être intégrées totalement ou partiellement. Les installations de programmation peuvent ainsi être réparties entre différents moyens externes et/ou internes par rapport à l'unité de commande contenant les zones de mémoire à programmer. Cela ne rend pas nécessaire une répartition entre les circuits câblés et les programmes qui constituent les moyens. Les moyens peuvent ainsi être réalisés ou copiés en circuits câblés et/ou en programmes. Une combinaison d'une mémoire ou d'une zone de mémoire et des données et/ou programmes localisés dans cette zone ou mémoire comme prescription de déroulement peut également être réalisée dans une logique câblée.

Comme indiqué, la programmation peut être lancée par un ordre d'effacement et elle se déroule alors automatiquement. L'opération de programmation peut ainsi être exécutée directement par le processeur CPU 101 de l'unité de commande. Un premier moyen peut être selon les indications ci-dessus, également un appareil de programmation externe 108 comme combinaison de circuits câblés de programme, comme une combinaison du processeur CPU 101 de l'unité de commande 100, une zone de mémoire SB 204 de la mémoire non fugitive 200 et des données et/ou programmes SDP associés à cette zone de mémoire ( $SDP_{anc}$  ou  $SDP_{nouv}$ ) ou des combinaisons comparables. Les moyens contiennent ainsi en plus une unité d'exécution comme par exemple le processeur CPU 101 ou le processeur existant dans l'appareil de programmation 108. La routine RAM, RR est enregistrée comme programme de copie dans la mémoire fugitive 206 ou dans sa zone de mémoire SB 207 et peut représenter ainsi un autre ou second moyen. D'autres moyens sont ainsi la zone de mémoire SB 201 avec le bloc Boot BB et la routine RAM, RR, localisée également prévue pour une coupure, la zone de mémoire SB 202 avec les programmes P1 ou la zone de mémoire SB 209 avec les programmes P2.

La figure 4 montre le cas selon lequel seule la zone de données SB 203 est programmée pour les données D. Un ordre correspondant lance également ici le déroulement des

opérations. Il peut s'agir là encore d'un ordre d'effacement mais qui montre que seulement la zone de données doit être programmée. Dans le bloc 401 on efface alors la zone de données SB 203. Dans le bloc 402 on programme la zone de données  
5. SB 203. On inscrit également les données Dnouv dans SB 203. Dans le bloc 403, on vérifie les données programmées comme dans l'exemple précédent. Si ainsi par l'interrogation 404 on décèle des défauts comme précédemment, on revient au bloc 401. Si la vérification est en ordre, on arrive au bloc 405,  
10 c'est-à-dire la fin du déroulement. Une simple programmation ou modification de données est de plus possible dans des conditions très simples.

R E V E N D I C A T I O N S

1°) Procédé pour modifier et/ou enregistrer des données et/ou des programmes, notamment des données et/ou des programmes de commande des opérations de fonctionnement dans un véhicule,  
5 dans au moins une mémoire non fugitive d'un système d'ordinateur, notamment une mémoire d'une unité de commande d'un véhicule, selon lequel, déjà dans la mémoire non fugitive, dans une zone originale, associée, on efface les anciennes données et/ou programmes avant de modifier et/ou d'enregistrer des  
10 données et/ou programmes nouveaux,

caractérisé en ce que

- les données et/ou programmes sont répartis en au moins des premières et des secondes données et/ou programmes, les secondes données ou programmes méritant d'être protégés ou  
15 contenant des données et/ou programmes méritant d'être protégés,
- les anciennes secondes données et/ou programmes ne sont d'abord effacés de la zone originale associée que lorsqu'on inscrit les nouvelles secondes données et/ou programmes dans une seconde zone de mémoire qui ne correspond pas à la zone originale, et  
20
- on inscrit un programme de copie dans une mémoire fugitive qui peut alors copier les nouvelles secondes données et/ou programmes de la seconde zone de mémoire dans la zone originale.
  
25

2°) Procédé pour modifier et/ou enregistrer les données et/ou des programmes, notamment des données et/ou des programmes de commande d'opération de fonctionnement d'un véhicule dans au moins une mémoire non fugitive d'un système d'ordinateur, notamment d'une mémoire d'une unité de commande d'un véhicule, selon lequel, déjà dans la mémoire non fugitive, on efface les anciennes données et/ou programmes contenus dans une zone de mémoire avant de modifier et/ou d'inscrire les nouvelles  
30 données et/ou programmes,

caractérisé en ce qu'

un programme de copie inscrit dans une mémoire fugitive enregistre les nouvelles données et/ou programmes dans la zone de

mémoire, après avoir effacé les anciennes données et/ou programmes, sans réaliser une copie de sécurité des anciennes données et/ou programmes.

5 3°) Procédé de modification et/ou d'enregistrement de données et/ou de programmes, notamment de données et/ou de programmes de commande des opérations de fonctionnement dans un véhicule, dans au moins une mémoire non fugitive d'un système d'ordinateur, notamment une mémoire d'une unité de commande  
10 du véhicule, selon lequel, déjà dans la mémoire non fugitive, dans les zones de mémoire associées, on efface les anciennes données et/ou programmes avant de modifier et/ou d'enregistrer de nouvelles données et/ou programmes au début du procédé avec un ordre d'effacement,

15 caractérisé en ce qu'  
on efface seulement partiellement les anciennes données et/ou programmes par l'ordre d'effacement, les données et/ou les programmes étant répartis en au moins des premières et des secondes données et/ou programmes, au moins les secondes données et/ou programmes méritant d'être protégés, ou contenant des données et/ou des programmes méritant d'être protégés, et les anciennes secondes données et/ou programmes dans la zone de mémoire associée restent conservés dans la zone originale.

25 4°) Procédé selon la revendication 1 ou 3,  
caractérisé en ce que  
les secondes données et/ou programmes ne sont exécutés que dans la zone originale.

30 5°) Procédé selon la revendication 1 ou 2,  
caractérisé en ce que  
le programme de copie est contenu dans les secondes données et/ou programmes et peut être chargé par ceux-ci dans la mémoire fugitive.

35 6°) Procédé selon la revendication 1 ou 2,  
caractérisé en ce que

le programme de copie est contenu dans des troisièmes données et/ou programmes qui peuvent exécuter une opération de démarrage du système d'ordinateur, et peut être chargé dans la mémoire fugitive par les troisièmes données et/ou programmes.

5

7°) Procédé selon la revendication 1 ou 3, caractérisé en ce que les secondes données et/ou programmes ne peuvent être exécutés que dans la zone originale.

10

8°) Dispositif notamment dans un véhicule automobile comportant au moins une mémoire non fugitive et au moins une mémoire fugitive ainsi que des premiers moyens pour modifier et/ou pour enregistrer des données et/ou des programmes dans au moins la mémoire non fugitive,

15 les premiers moyens effaçant déjà dans la mémoire non fugitive, dans une zone originale associée, les anciennes données et/ou programmes avant de modifier et/ou d'enregistrer les nouvelles données et/ou programmes,

20 caractérisé en ce qu'il comprend des seconds moyens qui contiennent la mémoire fugitive, les données et/ou programmes étant répartis en au moins des premières et des secondes données et/ou programmes, au moins les secondes données et/ou programmes méritant 25 d'être protégés ou contenant des données et/ou programmes méritant d'être protégés, et dans lequel les premiers moyens n'effaçent d'abord les anciennes secondes données et/ou programmes dans la zone originale associée de la mémoire non fugitive que si les premiers moyens ont inscrit les nouvelles 30 secondes données et/ou programmes dans une seconde zone de mémoire de la mémoire non fugitive, différente de la zone originale, et les seconds moyens peuvent copier les nouvelles secondes données et/ou programmes dans la zone originale.

35

9°) Dispositif notamment dans un véhicule comportant au moins une mémoire non fugitive et au moins une mémoire fugitive ainsi que des premiers moyens pour modifier et/ou pour enregistrer des données et/ou des programmes, notamment des don-

nées et/ou des programmes de commande d'opérations de fonctionnement dans le véhicule, dans la mémoire non fugitive et les premiers moyens effacent dans la mémoire non fugitive, les anciennes données et/ou programmes avant de modifier 5 et/ou d'enregistrer de nouvelles données et/ou programmes, caractérisé en ce qu'

il comprend des seconds moyens contenant la mémoire fugitive, qui inscrivent les nouvelles données et/ou programmes, après effacement des anciennes données et/ou programmes, dans la 10 mémoire non fugitive, sans que les premiers et les seconds moyens réalisent une copie de sécurité à partir des données anciennes et/ou programmes.

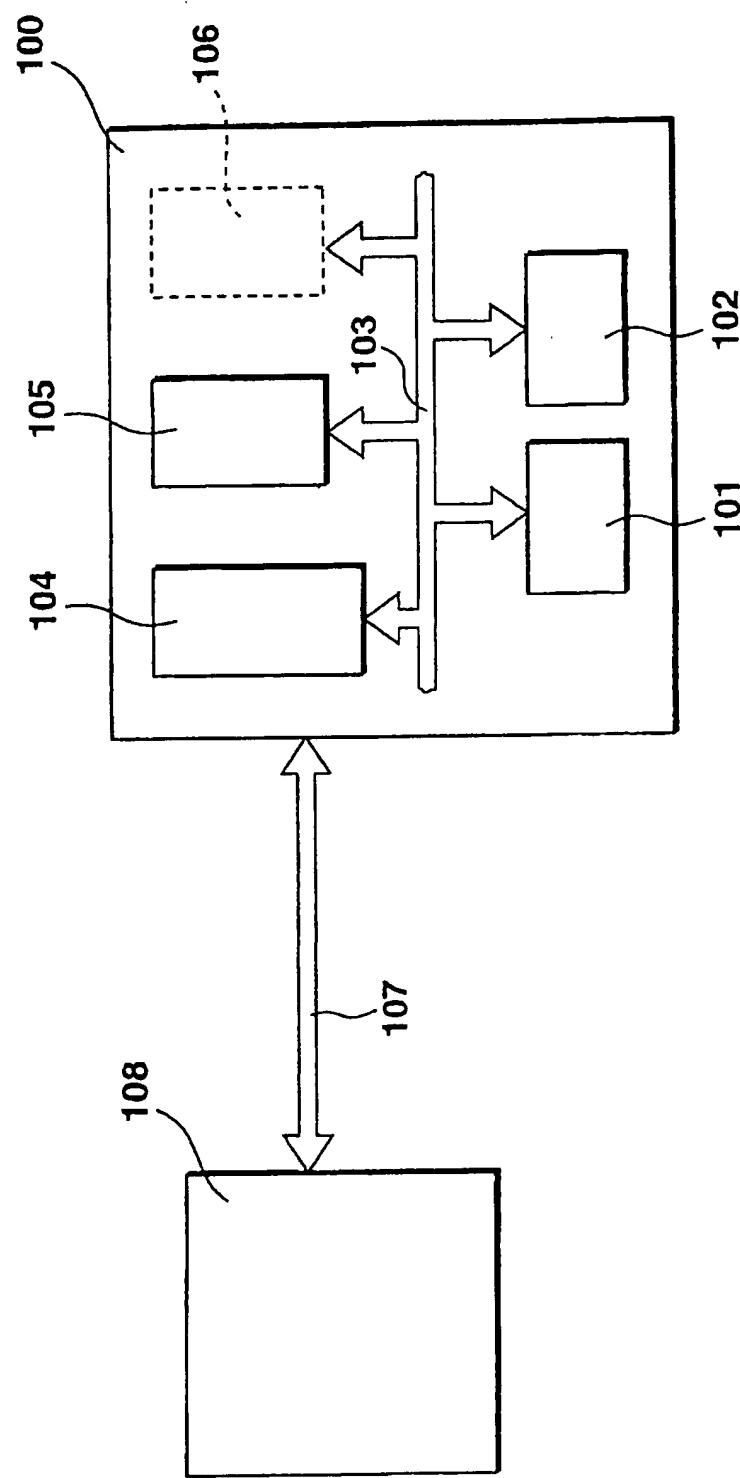
10°) Dispositif notamment pour un véhicule automobile, comportant au moins une mémoire non fugitive et au moins une mémoire fugitive ainsi que des premiers moyens pour modifier et/ou pour enregistrer des données et/ou des programmes dans au moins la mémoire non fugitive, en particulier des données et/ou des programmes de commande d'opérations de fonctionnement de véhicules automobiles, et les premiers moyens effacent déjà dans la première mémoire, dans des zones de mémoires associées, les anciennes données contenues et/ou programmes, avant de modifier et/ou d'enregistrer de nouvelles données et/ou programmes d'abord par un ordre d'effacement, 25

caractérisé en ce que

les anciennes données et/ou programmes ne sont effacés que partiellement par l'ordre d'effacement, les données et/ou les programmes étant répartis en au moins des premières et des 30 secondes données et/ou des programmes, au moins les secondes données et/ou programmes méritant d'être protégés ou contenant des données et/ou des programmes méritant d'être protégés, et ainsi des seconds moyens qui contiennent les anciennes secondes données et/ou programmes et la zone de mémoire associée, la zone originale, restent maintenus et les seconds moyens, après l'ordre d'effacement, exécutent la modification et/ou l'enregistrement de données et/ou de programmes. 35

1/4

FIGURE 1



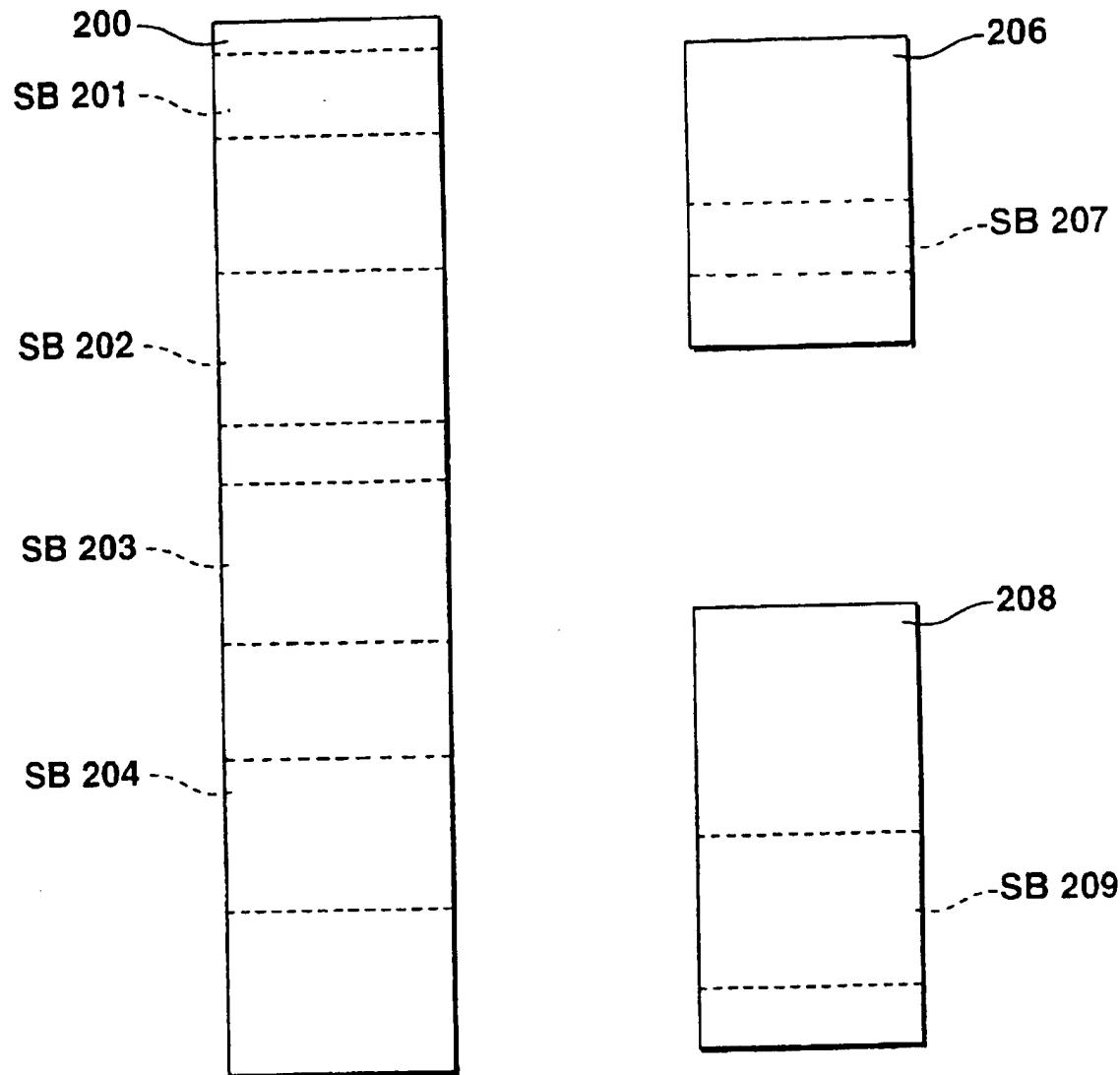


FIGURE 2

3/4

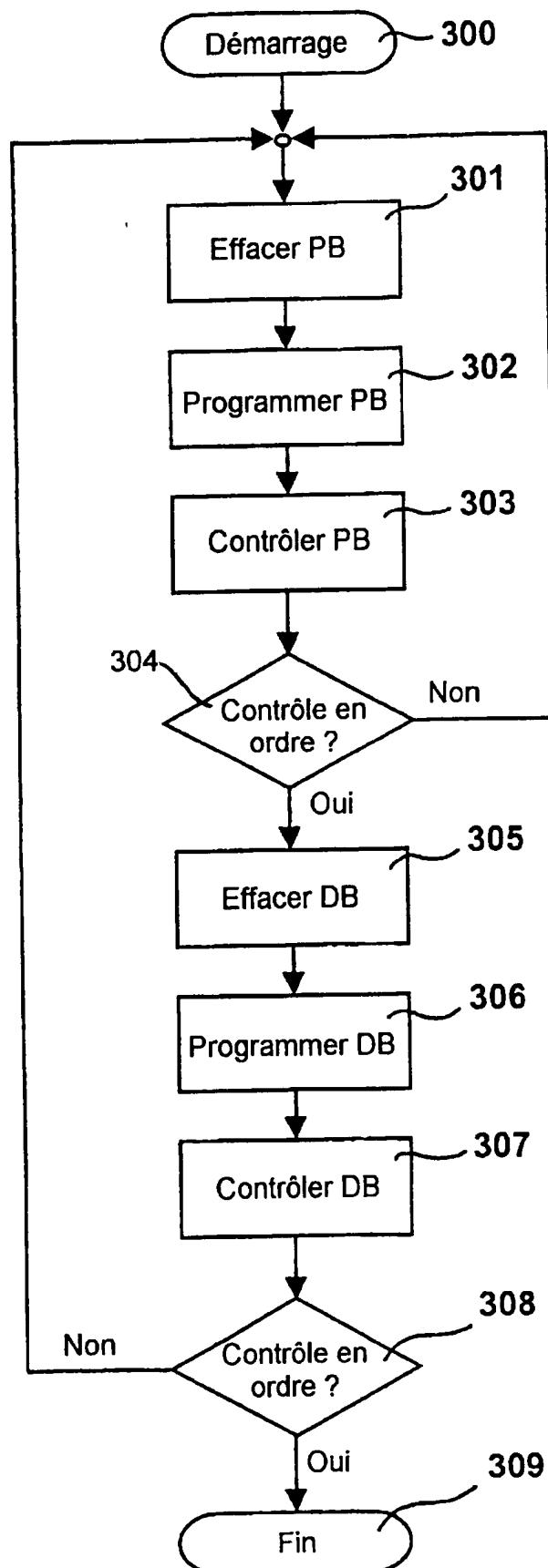


FIGURE 3

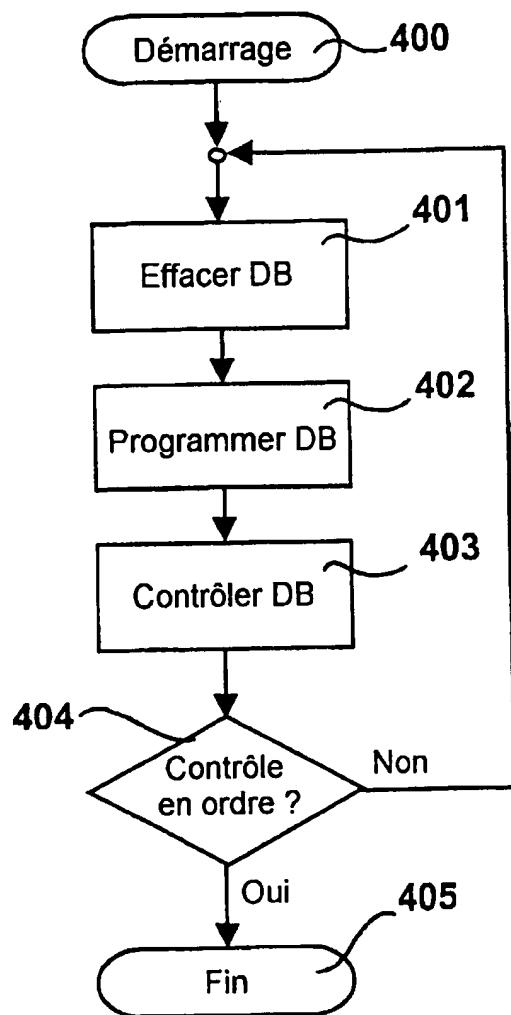


FIGURE 4